

## Certified Scrum Developer (CSD) Inhouse Training Leistungsbeschreibung



### Kursziel

**Ihr Nutzen:** Dieser Kurs bereitet Sie intensiv auf Ihre Arbeit als agiler Software-Entwickler vor. Sie lernen agile Entwicklungstechniken und Konzepte kennen und schaffen damit die Basis für qualitativ hochwertige Software und kontinuierliche Lieferfähigkeit. Das Training vermittelt den neuesten Stand agiler Softwareentwicklung wie testgetriebene Entwicklung, agiles Testen bis hin zu Continuous Integration und Delivery. Um den Praxistransfer sicherzustellen, erproben die Teilnehmer das Frischgelernte in der konkreten Zusammenarbeit als Scrum-Team.

### Zielgruppe

**Wer profitiert von diesem Kurs:** Der Kurs richtet sich an Softwareentwickler und Tester, die agile Entwicklungsmethoden lernen oder vertiefen möchten, oder diejenigen, die eine Zertifizierung zum Certified Scrum Developer anstreben. Voraussetzung für die Kursteilnahme ist Programmiererfahrung in Java. Weitere Programmiersprachen auf Anfrage.

### Die wichtigsten Inhalte auf einen Blick

Scrum-Teams konvertieren effektiv und verlässlich Anforderungen in fertige Software-Inkremente. Sie arbeiten produktiv mit dem Product Owner an der Detaillierung der Anforderungen. Die Teammitglieder wenden ihre Agile Engineering Skills pragmatisch an, um iterativ und inkrementell qualitativ hochwertige Software zu erstellen.

### Vorbereitung

Da der Kurs die Gelegenheit geben soll, praktische Probleme bei der Einführung von Scrum zu erörtern und Erfahrungen auszutauschen, empfehlen wir als Vorbereitung die Lektüre des [Scrum Guides](#) als offizielle Grundlage für Scrum.

Alle praktischen Übungen finden in Java mit JDK8 oder neuer statt. Als Entwicklungsumgebung (IDE) haben die Teilnehmer die Wahl zwischen Eclipse und IntelliJ, zum Versions- und Buildmanagement kommt git zum Einsatz. Die Teilnehmer stellen ihre eigenen Laptops und sollten vor Kursbeginn bereits Java und eine der beiden IDEs installiert haben.

## Lernumgebung und Qualität

Wir gestalten diesen Kurs sehr interaktiv in einer angenehmen und offenen Lernumgebung. Vertrauen und Partizipation sind uns dabei besonders wichtig. Die kurzweiligen Theorieblöcke wechseln sich mit extra hierfür konzipierten Übungen ab; Begriffe, Regeln und Abläufe werden dadurch erprobt und gefestigt, Scrum-Praktiken werden im Training konkret eingesetzt.

Der Kurs wird von zwei unserer sehr erfahrenen Trainern begleitet, die langjährige Erfahrung sowohl in der Softwareentwicklung als auch in Agile Coaching und Trainings haben.

## Zertifizierung

**So funktioniert die CSD-Zertifizierung:** Voraussetzung für die Zertifizierung als Certified Scrum Developer ist die Teilnahme an einem CSD-Kurs. Nach dem Kurs melden wir Sie bei der Scrum Alliance als neues Mitglied an.

Das CSD-Zertifikat gilt für zwei Jahre und kann danach erneuert werden. Weitere Informationen dazu erhalten Sie während des Kurses und auf der [Web-Seite der Scrum Alliance](#).



improv ist „Registered Education Provider“ der [Scrum Alliance](#).

## Format

Der Kurs kann entweder als 5-Tage-Block gebucht oder in zwei Einheiten à 2 Tage (Scrum und Agiles Testen) und 3 Tage (Agile Engineering Praktiken) aufgeteilt werden.

## SEUs und PDUs

Dieser Kurs wird von der Scrum Alliance mit 16 SEUs zur Verlängerung eines anderen Zertifikats bewertet, wenn die Teilnehmer bereits über ein anderes Zertifikat der Scrum Alliance wie bspw. den Certified Scrum Product Owner verfügen.

Sie können sich außerdem 16 PDUs beim [Project Management Institute](#) für diesen Kurs gutschreiben lassen.

## **Agenda Certified Scrum Developer**

Die Agenda entspricht den Learning Objectives der Scrum Alliance. Selbstverständlich können wir die Agenda im Rahmen dieser Vorgaben an Ihre spezifische Ausgangssituation anpassen.

<p><b>Tag 1: Scrum Kompakt</b> 8 Stunden inkl. Pausen (z.B. 9:00-17:00 Uhr)</p> <p><b>Einführung</b> Iterativ-inkrementeller Entwicklungsprozess</p> <p><b>Arbeiten in einem agilen Team</b> Grundlegende Teamfähigkeiten, Zusammenarbeit mit Product Owner und Kunden</p> <p><b>Grundlagen von Scrum</b> Rollen und Artefakte</p> <p><b>Anforderungs-Management und Release-Planung</b> Product Vision, Product Backlog und die Aufgaben des Product Owners, User Stories, Grobschätzung und Priorisierung, Velocity, Release-Planung</p> <p><b>Sprints und Meetings</b> Sprint Planning, Sprint-Review und -Retrospektive, Daily Scrums</p> <p><b>Scrum Klinik</b> Themen aus der praktischen Arbeit der Teilnehmer werden besprochen</p>	<p><b>Tag 2: Agile Engineering Skills I</b> 8 Stunden inkl. Pausen (z.B. 9:00-17:00 Uhr)</p> <p><b>Einführung Agile Engineering</b> Überblick Agile Engineering, Feedbackzyklen, Diskussion</p> <p><b>Test Driven Development (TDD)</b> TDD Zyklus (Red-Green-Refactor), Demonstration TDD, Struktur Unittests (arrange-act-assert), Coding Kata zu TDD</p> <p><b>Akzeptanztestgetriebene Entwicklung (ATDD)</b> ATDD Zyklus, Übung ATDD</p> <p><b>Praktische Übung an der Beispielanwendung</b> Sprint #1</p>
<p><b>Tag 3: Agile Engineering Skills II</b> 8 Stunden inkl. Pausen (z.B. 9:00-17:00 Uhr)</p> <p><b>Refactoring</b> Designverbesserung ohne Verhaltensänderung, Übung Refactoring</p> <p><b>Entkopplung und Mocking</b> Testen im komplexen Umfeld</p> <p><b>Praktische Übung an der Beispielanwendung</b> Sprint #2</p>	<p><b>Tag 4: Agile Engineering Skills III</b> 8 Stunden inkl. Pausen (z.B. 9:00-17:00 Uhr)</p> <p><b>Architektur</b></p> <p><b>Clean Code</b> Lesbarer und änderbarer Code, Übung Clean Code</p> <p><b>Continuous Integration &amp; Delivery</b></p> <p><b>Legacy Code</b> Arbeiten mit Legacy Code</p> <p><b>Praktische Übung an der Beispielanwendung</b> Sprint #3</p>

**Tag 5: Advanced Agile Testing**  
**8 Stunden inkl. Pausen (z.B. 9:00-17:00 Uhr)**

**Einführung**

Testen für Entwickler in agilen Projekten

**Automatisierte Tests**

Strategien, Organisation, Maintenance

**Test Smells und Clean Test Code**

**Advanced Unit Testing**

Die tatsächliche Reihenfolge der Themen kann von dieser Agenda abweichen.